

AMENDMENTS TO THE CLAIMS

Claims Pending:

- At time of the Action: Claims 1-38
- Amended Claims: Claims 1, 8, 17, and 27
- After this Response: Claims 1-38

1. (Currently Amended) A method, comprising:

receiving a request to add a new filter to a filter table stored in an inverse query engine cache;

adding the new filter to the filter table, wherein the new filter comprises at least one of a condition field, a data field, an expiration time field, a filter weight field, or and a permanent flag field;

determining the filter table of a bounded size;

maintaining the inverse query engine cache at or below a maximum cache size, wherein the size of the inverse query engine cache may be indicated by size of the filter table, estimate of size of the filter table, or by cache usage;

wherein the inverse query engine cache comprises at least one of an add filter module, a remove filter module, a matcher, a maintainer, a filter table, a most recently used list, or an expiration list;

wherein the expiration list comprises a filter identifier including an expiration value in an expiration field;

removing a filter based on an expiration time;

trimming the filter table; and

wherein the inverse query engine cache is used exclusively by an inverse query engine to store filters associated therewith.

2. (Original) The method as recited in claim 1, further comprising maintaining the size of the inverse query engine cache between an optimal cache size and the maximum cache size.

3. (Original) The method as recited in claim 1, wherein the maintaining further comprises:

determining if the addition of the new filter to the new filter table increases the cache size above the maximum cache size; and

removing one or more filters from the filter table if the addition of the new filter causes the cache size to exceed the maximum cache size.

4. (Original) The method as recited in claim 3, wherein the determining step further comprises:

determining a relative size of the new filter;

assigning a filter weight to the new filter based on the relative filter size;

deriving a cache weight by summing filter weights of all filters in the filter table including the new filter; and

comparing the cache weight to the maximum cache size.

5. (Original) The method as recited in claim 1, wherein the maintaining further

comprises:

- identifying a weight associated with the new filter;
- adding the weight associated with the new filter to a cache weight that is the sum of filter weights of filters in the filter table, each filter having a filter weight; and
- comparing the cache weight to the maximum cache size.

6. (Original) The method as recited in claim 1, further comprising identifying one or more expired filters in the filter table; and wherein the maintaining the inverse query engine cache further comprises removing at least one of the identified expired filters.

7. (Original) The method as recited in claim 1, further comprising a least recently used filter in the filter table; and wherein the maintaining the inverse query engine cache further comprises removing the least recently used filter from the filter table when a size of the inverse query engine cache reaches the maximum cache size.

8. (Currently Amended) A system, comprising:

- an inverse query engine configured to test an input against a collection of filters;
- cache associated with the inverse query engine, wherein the inverse query engine cache comprises at least one of an add filter module, a remove filter module, a matcher, a maintainer, a filter table, a most recently used list, or an expiration list;

a the filter table stored in the cache and containing multiple filters, wherein the new filter comprises ~~at least one of~~ a condition field, a data field, an expiration time field, a filter weight field, ~~or~~ and a permanent flag field; and

a maintainer configured to maintain a size of the filter table within definite cache bounds, wherein the maintainer removes a filter based on an expiration time from the cache and trims the cache;

wherein the size of the filter table may be indicated by size of the filter or by weight of the filter.

9. (Original) The system as recited in claim 8, further comprising an expiration module configured to remove expired filters from the filter table.

10. (Original) The system as recited in claim 9, wherein the maintainer is further configured to invoke the expiration module when a new filter is added to the filter table.

11. (Original) The system as recited in claim 8, further comprising a trim module configured to remove one or more filters from the filter table when the cache reaches a maximum cache size.

12. (Original) The system as recited in claim 11, wherein the trim module is further configured to remove the one or more filters from the filter table until the cache is reduced to an optimal cache size.

13. (Original) The system as recited in claim 11, wherein the trim module is further configured to determine if a permanent flag in a filter is set and, if the permanent flag is set, to leave the filter in the filter table.

14. (Original) The system as recited in claim 11, wherein:
a filter weight is associated with each filter in the filter table;
the cache size further comprises a cache weight that is a sum of all filter weights in the filter table;
the maximum cache size further comprises a maximum cache weight; and
wherein the trim module is further configured to deduct a filter weight from the cache weight when a filter associated with the filter weight is removed from the filter table.

15. (Original) The system as recited in claim 14, further comprising a cache weight module configured to assign a filter weight to each filter added to the filter table, each filter weight identifying a relative size of a filter with regard to other filters in the filter table

16. (Original) The system as recited in claim 11, further comprising:
a most recently used list configured to track usage of filter table filters according to when filters are used, and indicating that a filter has been most recently used when the filter is the latest filter to match an input; and

wherein the trim module is further configured to remove the one or more filters from the filter table based on which filters have been least recently used.

17. (Currently Amended) One or more computer-readable storage media storing computer-executable instructions that, when executed on a computer, perform the following steps:

receiving a request to add a new query to an inverse query engine cache that stores multiple queries, each query having a query size associated therewith;

defining conditions and processing input that satisfies the conditions;

deriving a cache size that is a sum of query sizes of the queries stored in the inverse query engine;

determining if the cache size is at greater than or equal to a maximum cache size, wherein the cache size may be determined comprising cache usage, size of the query, or estimate of size of the query;

removing one or more queries from the inverse query engine cache if the cache size is greater than or equal to the maximum cache size;

deducting the query size of each query removed from the cache size;

adding the new query to the inverse query engine cache; and

adding a new query size to the cache size, the new query size identifying a size of the new query added to the inverse query engine cache.

18. (Original) The one or more computer-readable media as recited in claim 17, wherein the removing step further comprises removing a query from the inverse query

engine that has been used less recently than other queries stored in the inverse query engine cache.

19. (Original) The one or more computer-readable media as recited in claim 17, wherein a query size is an estimate of the actual size of the query.

20. (Original) The one or more computer-readable media as recited in claim 17, wherein the step of adding the new query size to the cache size is performed before determining if the cache size is greater than or equal to the maximum cache size.

21. (Original) The one or more computer-readable media as recited in claim 17, further comprising instructions to perform the following additional steps:

determining if any queries stored in the inverse query engine cache have expired;
and

removing one or more expired queries from the inverse query engine cache.

22. (Original) The one or more computer-readable media as recited in claim 17, wherein the new query size is received with the new query.

23. (Original) The one or more computer-readable media as recited in claim 17, further comprising instructions to perform the additional step of determining the new query size.

24. (Original) The one or more computer-readable media as recited in claim 23, wherein the determining the new query size further comprises estimating the new query size.

25. (Original) The one or more computer-readable media as recited in claim 17, further comprising the additional step of:

determining if a query in the inverse query engine cache includes an indication that the query is a permanent query; and

wherein the query is not removed from the inverse query engine cache if the query is a permanent query.

26. (Original) The one or more computer-readable media as recited in claim 17, wherein a query size is represented as a weight value that denotes the relative size of the query associated therewith with regard to other queries stored in the inverse query engine cache.

27. (Currently Amended) A method for maintaining an inverse query engine cache, comprising:

determining when inverse query engine cache usage is approaching a cache usage capacity, wherein the cache usage capacity is determined by a size of a filter table comprising ~~at least one of~~ a condition field, a data field, an expiration time field, a filter weight field, ~~or~~ and a permanent flag field; and

removing one or more filters from the inverse query engine cache when the cache is approaching the cache capacity until the cache usage is reduced to an optimal cache usage,

wherein removing one or more filters comprises at least one of expiring or trimming the cache; and

wherein an inverse query engine cache comprises at least one of an add filter module, a remove filter module, a matcher, a maintainer, a filter table, a most recently used list, or an expiration list.

28. (Original) The method as recited in claim 27, wherein the removing step further comprises removing at least one expired filter from the inverse query engine cache.

29. (Original) The method as recited in claim 28, wherein an expired filter is a filter having an expiration time associated therewith that precedes a current time.

30. (Original) The method as recited in claim 28, wherein an expired filter is a filter that has been stored in the inverse query engine cache for a predefined period of time.

31. (Original) The method as recited in claim 27, wherein the removing step further comprises removing a least recently used filter from the inverse query engine cache.

32. (Original) An inverse query engine having an integrated cache.

33. (Original) The inverse query engine as recited in claim 32, wherein the cache is bound to a finite size.

34. (Original) The inverse query engine as recited in claim 32, wherein the cache is maintained within predefined limits by removing expired filters from a set of filters stored in the cache.

35. (Original) The inverse query engine as recited in claim 34, wherein an expired filter is a filter having an expiration time associated therewith that is earlier than a current time.

36. (Original) The inverse query engine as recited in claim 34, wherein an expired filter is a filter that has been stored in the cache for at least a specified period of time.

37. (Original) The inverse query engine as recited in claim 32, wherein the cache is maintained within predefined limits by removing at least one filter from a group of filters stored in the cache that has been used less recently than other filters in the group of filters.

38. (Original) The inverse query engine as recited in claim 37, wherein the at least one filter is removed only if the filter does not have an indication associated therewith that identifies the filter as a permanent filter that is not to be removed from the cache.